An Asymptotically Optimal On-Line Algorithm for Parallel Machine Scheduling

Mabel Chou
Singapore-MIT Alliance
Department of Decision Sciences
National University of Singapore
Email: bizchoum@nus.edu.sg

Maurice Queyranne
University of British Columbia
Vancouver B.C., Canada V6T 1Z2

David Simchi-Levi
Massachusetts Institute of Technology
Cambridge MA 02139,USA

Abstract — Jobs arriving over time must be non-preemptively processed on one of m parallel machines, each of which running at its own speed, so as to minimize a weighted sum of the job completion times. In this on-line environment, the processing requirement and weight of a job are not known before the job arrives. The Weighted Shortest Processing Requirement (WSPR) on-line heuristic is a simple extension of the well known WSPT heuristic, which is optimal for the single machine problem without release dates. We prove that the WSPR heuristic is asymptotically optimal for all instances with bounded job processing requirements and weights. This implies that the WSPR algorithm generates a solution whose relative error approaches zero as the number of jobs increases. Our proof does not require any probabilistic assumption on the job parameters and relies extensively on properties of optimal solutions to a single machine relaxation of the problem.

I. Introduction

In the uniform parallel machine minsum scheduling problem with release dates, jobs arrive over time and must be allocated for processing to one of m given parallel machines. Machine M_i $(i = 1, \ldots, m)$ has speed $s_i > 0$ and can process at most one job at a time. Let n denote the total number of jobs to be processed and let $N = \{1, 2, \ldots, n\}$. Job $j \in N$ has processing requirement $p_j \geq 0$, weight $w_j > 0$, and release date $r_j \geq 0$. The processing of job j cannot start before its release date r_j and cannot be interrupted once started on a machine. If job j starts processing at time S_j on machine M_i, then it is completed p_j/s_i time units later; that is, its completion time is $C_j = S_j + p_j/s_i$. In the single machine case, i.e., when $m = 1$, we may assume that $s_1 = 1$ and in this case the processing requirement of a job is also referred to as the job processing time.

We seek a feasible schedule of all n jobs, which minimizes the minsum objective $\sum_{j=1}^{n} w_j C_j$, the weighted sum of completion times. In standard scheduling notation, see, e.g., [5], this problem is denoted $Q|r_j|\sum w_j C_j$. Our main result concerns the case where the set of m parallel machines is held fixed, which is usually denoted as problem $Qm|r_j|\sum w_j C_j$.

In practice, the precise processing requirement, weight and release date (or arrival time) of a job may not be known before the job actually arrives for processing. We consider an on-line environment where these data p_j, w_j and r_j are not known before time r_j. Thus scheduling decisions have to be made over time, using at any time only information about the jobs already released by that time; see, e.g., [10] for a survey of on-line scheduling.

In a competitive analysis, we compare the objective value, $Z^A(I)$, of the schedule obtained by applying a given (deterministic) on-line algorithm A to an instance I to the optimum (off-line) objective value $Z^*(I)$ of this instance. The competitive ratio of algorithm A, relative to a class $\mathcal{I}$ of instances (such that $Z^*(I) > 0$ for all $I \in \mathcal{I}$), is

$$c_{\mathcal{I}}(A) = \sup \left\{ \frac{Z^A(I)}{Z^*(I)} : I \in \mathcal{I} \right\} .$$

One of the best known results [11] in minsum scheduling is that the single machine problem without release dates, $1||\sum w_j C_j$, is solved to optimality by the following Weighted Shortest Processing Time (WSPT) algorithm: process the jobs in nonincreasing order of their weight-to-processing time ratio w_j/p_j. Thus the competitive ratio $c_{\mathcal{I}}(WSPT) = 1$ for the class $\mathcal{I}$ of all instances of the single machine problem $1||\sum w_j C_j$. Unfortunately, this result does not extend to problems with release dates or with parallel machines; in fact the single machine problem with release dates and equal weights, $1|r_j|\sum C_j$, and the identical parallel machine problem, $P||\sum w_j C_j$ are already NP-hard [8]. Consequently, a great deal of work has been devoted to the development and analysis of heuristics, in particular, Linear Programming (LP) based heuristics, with attractive competitive ratios.

A departure from this line of research was presented in [6]. To present their results, define the asymptotic performance ratio $R_{\mathcal{I}}^{\infty}(A)$ of an algorithm A, relative to instance class $\mathcal{I}$, as

$$\inf \left\{ r \geq 1 \mid \exists n_0 \text{ s.t. } \frac{Z^A(I)}{Z^*(I)} \leq r, \forall I \in \mathcal{I} \text{ with } n \geq n_0 \right\} .$$

Thus, the asymptotic performance ratio characterizes the maximum relative deviation from optimality for all "sufficiently large" instances in $\mathcal{I}$. When A is an on-line algorithm, and $Z^*(I)$ still denotes the off-line optimum objective value, we call $R_I^\infty(A)$ the asymptotic competitive ratio of A relative to instance class I.

[6] focused on the single machine total completion time problem with release dates, i.e., problem $1|r_j|\sum C_j$ and analyzed the effectiveness of a simple on-line dispatch rule, referred to as the Shortest Processing Time among Available jobs (SPTA) heuristic. In this algorithm, at the completion time of any job, one considers all the jobs which have been released by that date but not yet processed, and select the job with the smallest processing time to be processed next. If no job is available the machine is idle until at least one job arrives. Results in [6] imply that the asymptotic competitive ratio of the SPTA heuristic is equal to one for all classes of instances with bounded processing times, i.e., instance classes I for which there exist constants $\bar{p} \geq \underline{p} \geq 0$ such that $\underline{p} \leq p_j \leq \bar{p}$ for all jobs j in every instance $I \in I$.

It is natural to try and extend the WSPT and SPTA heuristics to problems with release dates and/or parallel machines. A simple extension to the problem $Q|r_j|\sum w_j C_j$ considered herein, with both uniform parallel machines and release dates, is the following WSPR algorithm: whenever a machine becomes idle, start processing on it an available job, if any, with largest w_j/p_j ratio; otherwise, wait until the next job release date. This is a very simple on-line algorithm which is fairly myopic and clearly suboptimal, for at least two reasons. First, it is "non-idling", that is, it keeps the machines busy so long as there is work available for processing; this may be suboptimal if a job k with large weight w_k (or short processing requirement p_k) is released shortly thereafter and is forced to wait because all machines are then busy. Second, for machines with different speeds, the WSPR algorithm arbitrarily assigns jobs to the idle machines, irrespective of their speeds; thus an important job may be assigned to a slow machine while a faster machine is currently idle or may become idle soon thereafter. Thus it is easy to construct instance classes I, for example with two jobs on a single machine, for which the WSPR heuristic performs very poorly; this implies that its competitive ratio $c_I(WSPR)$ is unbounded.

In contrast, the main result of this paper is that the asymptotic competitive ratio $R_I^\infty(WSPR)$ of the WSPR heuristic is equal to one for all classes I of instances with a fixed set of machines and with bounded job weights and processing requirements.

Formally, our main result is presented in the following theorem.

Theorem 1: Consider any class I of instances of the uniform parallel machines problem $Qm|r_j|\sum w_j C_j$ with a fixed set of m machines, and with bounded weights and processing times, that is, for which there exist constants $\bar{w} \geq \underline{w} > 0$ and $\bar{p} \geq \underline{p} > 0$ such that $\underline{w} \leq w_j \leq \bar{w}$ and $\underline{p} \leq p_j \leq \bar{p}$ for all jobs j in every instance $I \in I$: Then the asymptotic competitive ratio of the WSPR heuristic is $R_I^\infty(WSPR) = 1$ for instance class I.

To put our results in perspective, it is appropriate at this point to refer to the work of Uma and Wein [12] who perform extensive computational studies with various heuristics including WSPR as well as linear programming based approximation algorithms for the single machine problem $1|r_j|\sum w_j C_j$. While Uma and Wein note that it is trivial to see that the worst-case performance of the WSPR heuristic is unbounded, they find that, on most data sets they used, this heuristic is superior to all the LP relaxation based approaches. The results in the present paper provide a nice explanation of this striking behavior reported by Uma and Wein. Indeed, our results show that if the job parameters, i.e., weights and processing times, are bounded, then the WSPR algorithm generates a solution whose relative error decreases to zero as the number of jobs increases. Put differently, WSPR has an unbounded worst-case performance only when the job parameters are unbounded.

II. A Mean Busy Date Relaxation for Uniform Parallel Machines

Let $N = \{1, \ldots, n\}$ be a set of jobs to be processed, with a given vector $p = (p_1, \ldots, p_n)$ of job processing requirements. Given any (preemptive) schedule we associate with each job $j \in N$ its processing speed function σ_j, defined as follows: for every date t we let $\sigma_j(t)$ denote the speed at which job j is being processed at date t. For example, for uniform parallel machines, $\sigma_j(t) = s_{i(j,t)}$ is the speed of the machine $M_{i(j,t)}$ processing job j at date t, and $\sigma_j(t) = 0$ if job j is idle at that date. Thus, for a single machine with unit speed, $\sigma_j(t) = 1$ if the machine is processing job j at date t, and 0 otherwise. We consider schedules that are complete in the following sense. First we assume that $0 \leq \sigma_j(t) \leq \bar{s}$ for all j and t, where $\bar{s}$ is a given upper bound on the maximum speed at which a job may be processed. Next, we assume that all processing occurs during a finite time interval $[0, T]$, where T is a given upper bound on the latest job completion time in a schedule under consideration. For the single machine problem we may use $\bar{s} = 1$ and for both the WSPR and LP schedules, $T = \max_j r_j + \sum_j p_j$. For the uniform parallel machines problem, we may use $\bar{s} = \max_i s_i$ and $T = \max_j r_j + \sum_j p_j / \min_i s_i$. The assumption

$$\int_0^T \sigma_j(\tau)\, d\tau = p_j$$

then express the requirement that, in a complete schedule, each job is entirely processed during this time interval $[0, T]$. The preceding assumptions imply that all integrals below are well defined.

The mean busy date M_j of job j in a complete schedule is the average date at which job j is being processed, that is,

$$M_j = \frac{1}{p_j} \int_0^T \sigma_j(\tau)\, \tau\, d\tau\,.$$

We let $M = (M_1, \ldots, M_n)$ denote the mean busy date vector, or MBD vector, of the schedule. When the speed function σ_j is piecewise constant, we may express the mean busy date M_j as the weighted average of the midpoints of the time intervals during which job j is processed at constant speed, using as weights the fraction of its work requirement p_j processed in these intervals. Namely, if $\sigma_j(t) = s_{j,k}$ for $a_k < t < b_k$, with $0 \le a_1 < b_1 < \ldots < a_K < b_K \le T$ and $\sum_{k=1}^{K} s_{j,k}(b_k - a_k) = p_j$ then

$$M_j = \sum_{k=1}^{K} \frac{s_{j,k}(b_k - a_k)}{p_j} \, \frac{a_k + b_k}{2}. \qquad (1)$$

Thus, if job j is processed without preemption at speed $s_{j,1}$, then its completion time is $C_j = M_j + \frac{1}{2} p_j / s_{j,1}$. In any complete schedule, the completion time of every job j satisfies $C_j \ge M_j + \frac{1}{2} p_j / \bar{s}$, with equality if and only if job j is processed without preemption at maximum speed $\bar{s}$.

Let $w = (w_1, \ldots, w_n)$ denote the vector of given job weights. We use scalar product notation, and let $w^\top p = \sum_j w_j p_j$ and $w^\top C = \sum_j w_j C_j$, the latter denoting the minsum objective of instance I of the scheduling problem under consideration. We call $w^\top M = \sum_j w_j M_j$ the mean busy date objective, or MBD objective, and the problem $Z^{P-MBD}(I) = \min\{w^\top M : M$ is the MBD vector of a feasible preemptive schedule for instance $I\}$ the preemptive MBD problem. Letting $Z^*(I)$ denote the optimum minsum objective value $w^\top C$ of a feasible nonpreemptive schedule, it follows from $w \ge 0$ and the preceding observations that $Z^*(I) \ge Z^{P-MBD}(I) + (1/2\bar{s}) w^\top p$. Accordingly, we shall also refer to the preemptive MBD problem as the preemptive MBD relaxation of the original nonpreemptive minsum scheduling problem.

The preemptive MBD problem is well solved, see [3] and [4], for the case of a single machine with constant speed $s > 0$ and job release dates:

Theorem 2 ((Goemans)): The LP schedule defines an optimal solution to the preemptive MBD problem $1|r_j; pmtn|\sum w_j M_j$.

Let C^{LP} and M^{LP} denote the completion time vector, resp., the MBD vector, of the LP schedule, and let $Z^{MBD}(I) = Z^{P-MBD}(I) + (1/2\bar{s}) w^\top p$. Theorem 2 implies that $Z^*(I) \ge Z^{MBD}(I) = w^\top M^{LP} + (1/2\bar{s}) w^\top p$.

We will bound the maximum delay that certain amounts of "work" can incur in the WSPR schedule, relative to the LP schedule. For this, we now present a decomposition of the MBD objective $w^\top M$ using certain "nested" job subsets. Some of the results below were introduced in [3] in the context of single machine scheduling. We consider a general scheduling environment and a complete schedule, as defined towards the beginning of this Section. Assume, without loss of generality, that the jobs are indexed in a nonincreasing order of their ratios of weight to processing requirement:

$$w_1/p_1 \ge w_2/p_2 \ge \ldots \ge w_n/p_n \ge w_{n+1}/p_{n+1} = 0. \qquad (2)$$

Accordingly, job k has lower WSPR priority than job j if and only if $k > j$. In case of ties in (2), we consider WSPR priorities and an LP schedule which are consistent with the WSPR schedule. For $h = 1, \ldots, n$, let $\Delta_h = w_h/p_h - w_{h+1}/p_{h+1}$, and let $[h] = \{1, 2, \ldots, h\}$ denote the set of the h jobs with highest priority. For any feasible (preemptive) schedule we have: $w^\top M = \sum_{j=1}^{n} \frac{w_j}{p_j} p_j M_j = \sum_{j=1}^{n} \sum_{k=j}^{n} \Delta_k \, p_j M_j = \sum_{h=1}^{n} \Delta_h \sum_{j \in [h]} p_j M_j$. For any subset $S \subseteq N = \{1, \ldots, n\}$, let $p(S) = \sum_{j \in S} p_j$ denote its total processing time, and let $\sigma_S = \sum_{j \in S} \sigma_j$ denote its processing speed function. Define its mean busy date $M_S = (1/p(S)) \int_0^T \sigma_S(\tau)\, \tau \, d\tau$. Note that, in a feasible schedule, $\int_0^T \sigma_S(\tau)\, d\tau = p(S)$ and $\sum_{j \in S} p_j M_j = \int_0^T \sigma_S(\tau)\, \tau \, d\tau = p(S) M_S$. Therefore, we obtain the MBD objective decomposition

$$w^\top M = \sum_{h=1}^{n} \Delta_h \, p([h]) \, M_{[h]}. \qquad (3)$$

This decomposition allows us to concentrate on the mean busy dates $M_{[h]}$ of the job subsets $[h]$ $(h = 1, \ldots, n)$.

For any date $t \le T$, let $R_S(t) = \int_t^T \sigma_S(\tau)\, d\tau = p(S) - \int_0^t \sigma_S(\tau)\, d\tau$ denote the unprocessed work from set S at date t. (Note that this unprocessed work may include the processing time of jobs not yet released at date t.) Since the unprocessed work function $R_S(t)$ is nonincreasing with time t, we may define its (functional) inverse R_S^{-1} as follows: for $0 \le q \le p(S)$ let $R_S^{-1}(q) = \inf\{t \ge 0 : R_S(t) \le q\}$. Thus the processing date $R_S^{-1}(q)$ is the earliest date at which $p(S) - q$ units of work from set S have been processed. For any feasible schedule with a finite number of preemptions we have

$$\int_0^T R_S(t)\, dt = \int_0^{p(S)} R_S^{-1}(q)\, dq.$$

The mean busy date M_S can be expressed using the processing date function R_S^{-1}:

$$p(S) M_S = \int_0^T \sigma_S(t)\, t \, dt = \int_0^T \int_0^t \sigma_S(t)\, d\tau \, dt$$

$$= \int_0^T \int_\tau^T \sigma_S(t)\, dt \, d\tau = \int_0^T R_S(\tau)\, d\tau = \int_0^{p(S)} R_S^{-1}(q)\, dq.$$

Combining this with equation (3) allows us to express the MBD objective using the processing date function:

$$w^\top M = \sum_{h=1}^{n} \Delta_h \int_0^{p([h])} R_{[h]}^{-1}(q)\, dq. \qquad (4)$$

We now present a mean busy date relaxation for uniform parallel machines and then use the above expression (4) to bound the difference between the minsum objectives of the WSPR and LP schedules.

Assume that we have m parallel machines $\mathcal{M}_1, \ldots, \mathcal{M}_m$, where machine $\mathcal{M}_i$ has speed $s_i > 0$. Job j has processing requirement $p_j > 0$; if it is processed on machine $\mathcal{M}_i$ then its actual processing time is $p_{ij} = p_j / s_i$. We assume that the set of machines and their speeds are fixed and, without loss of generality, that the machines are indexed in nonincreasing order of their speeds, that is, $s_1 \ge s_2 \ge \dots \ge s_m > 0$. We have job release dates $r_j \ge 0$ and weights $w_j \ge 0$, and we seek a nonpreemptive feasible schedule in which no job j is processed before its release date, and which minimizes the minsum objective $\sum_j w_j C_j$. Since the set of m parallel machines is fixed, this problem is usually denoted as $Qm|r_j|\sum w_j C_j$. First, we present a fairly natural single machine preemptive relaxation, using a machine with speed $s^{[m]} = \sum_{i=1}^{m} s_i$. We then compare the processing date functions for the high WSPR priority sets $[h]$ between the WSPR schedule on the parallel machines and LP schedule on the speed-$s^{[m]}$ machine. We show an $O(n)$ additive error bound for the WSPR heuristic, for every instance class with a fixed set of machines and bounded job processing times and weights. This implies the asymptotic optimality of the WSPR heuristic for such classes of instances.

Consider any feasible preemptive schedule on the parallel machines, with completion time vector C. Recall that $\sigma_j(t)$ denotes the speed at which job j is being processed at date t in the schedule, and that the mean busy date M_j of job j is $M_j = (1/p_j)\int_0^T \sigma_j(\tau)\,\tau\,d\tau$ (where T is an upper bound on the maximum job completion time in any schedule being considered).

To every instance I of the uniform parallel machines problem we associate an instance $I^{[m]}$ of the single machine preemptive problem with the same job set N and in which each job $j \in N$ now has processing time $p_j^{[m]} = p_j/s^{[m]}$. The job weights w_j and release dates r_j are unchanged. Thus we have replaced the m machines with speeds $s_1, \dots, s_m$ with a single machine with speed $s^{[m]} = \sum_{i=1}^{m} s_i$. Consider any feasible preemptive schedule for this single machine problem and let $C^{[m]}$ denote its completion time vector. Let $I_j^{[m]}(t)$ denote the speed (either $s^{[m]}$ or 0) at which job j is being processed at date t. Thus the mean busy date $M_j^{[m]}$ of job j for this single machine problem is $M_j^{[m]} = (1/p_j)\int_0^T I_j^{[m]}(\tau)\,\tau\,d\tau$.

In the following Lemma, the resulting inequality $C_j^{[m]} \le C_j$ on all job completion times extends earlier results of [1] for the case of identical parallel machines (whereby all $s_i = 1$), and of [9] for a broad class of shop scheduling problems (with precedence delays but without parallel machines). To our knowledge, the mean busy date result, $M_j^{[m]} = M_j$, which we use later on, is new.

Lemma 1: To every feasible (preemptive or nonpreemptive) schedule with a finite number of preemptions[1] and with mean busy date vector M and completion time vector C on the uniform parallel machines, we can associate a feasible preemptive schedule with mean busy date vector $M^{[m]}$ and completion time vector $C^{[m]}$ on the speed-$s^{[m]}$ machine, such that $M_j^{[m]} = M_j$ and $C_j^{[m]} \le C_j$ for all jobs $j \in N$.

Proof: Let S_j denote the start date of job j in the given parallel machines schedule. Partition the time interval $[\min_j S_j, \max_j C_j]$ into intervals $[a_{t-1}, a_t]$ $(t = 1, \dots, \tau)$ such that exactly the same jobs are being processed by exactly the same machines throughout each interval. Thus $\{a_t : t = 0, \dots, \tau\}$ is the set of all job start dates and completion times, and all dates at which some job is preempted. Partition each job j into τ pieces (j, t) with work amount $q_{jt} = s_{i(j,t)}(a_t - a_{t-1})$ if job j is being processed during interval $[a_{t-1}, a_t]$ on a machine $M_{i(j,t)}$, and zero otherwise. Since each job j is performed in the given schedule, its processing requirement is $p_j = \sum_{t=1}^{\tau} q_{jt}$. Since each machine processes at most one job during interval $[a_{t-1}, a_t]$, we have $\sum_{j \in N} q_{jt} \le s^{[m]}(a_t - a_{t-1})$ for all t, with equality iff no machine is idle during interval $[a_{t-1}, a_t]$. Therefore the speed-$s^{[m]}$ machine has enough capacity to process all the work $\sum_{j \in N} q_{jt}$ during this interval. Construct a preemptive schedule on the speed-$s^{[m]}$ machine as follows. For each $t = 1, \dots, \tau$, fix an arbitrary sequence $(j_1, t), \dots, (j_{n(t)}, t)$ of the $n(t)$ pieces (j, t) with $q_{jt} > 0$. Starting at date a_{t-1} process half of each such piece (j, t) (i.e., for $\frac{1}{2}q_{jt}/s^{[m]}$ time units) in the given sequence. This processing is complete no later than date $\mu_t = \frac{1}{2}(a_{t-1} + a_t)$, the midpoint of the interval $[a_{t-1}, a_t]$. Then "mirror" this partial schedule about this midpoint μ_t by processing the other half of each piece in reverse sequence so as to complete this mirrored partial schedule precisely at date a_t. Since no job starts before its release date, all the processing requirement of every job is processed, and the speed-$s^{[m]}$ machine processes at most one job at a time, the resulting preemptive schedule is indeed feasible. Furthermore each job j completes at the latest at date $\max\{a_t : q_{jt} > 0\} = C_j$, so $C_j^{[m]} \le C_j$. Finally, the "mirroring" applied in each interval $[a_{t-1}, a_t]$ ensures that, for all jobs $j \in N$

$$\int_{a_{t-1}}^{a_t} \sigma_j^{[m]}(\tau)\tau\,d\tau = \frac{q_{jt}}{s^{[m]}} s^{[m]} \mu_t = \frac{q_{jt}}{s_i} s_i \mu_t = \int_{a_{t-1}}^{a_t} \sigma_j(\tau)\tau\,d\tau$$

where s_i is the speed of the machine M_i on which job j is processed during interval $[a_{t-1}, a_t]$ in the given parallel machines schedule. Adding over all intervals implies $M_j^{[m]} = M_j$ for all jobs $j \in N$. The proof is complete. ∎

Lemma 1 implies that the preemptive single machine problem, with a speed-$s^{[m]}$ machine, is a relaxation of the original uniform parallel machines problem, for any objective function (including the minsum objective with all $w_j \ge 0$) which is nondecreasing in the job completion times. For the minsum objective $\sum_j w_j C_j$, we may combine this result with Theorem 2 and obtain:

Corollary 1: Let $Z^*(I)$ denote the optimum objective value for instance I of the parallel machines problem $Q|r_j|\sum w_j C_j$. Let $M^{LP[m]}$ denote the mean busy date vector of the LP schedule for the corresponding instance

[1]The finiteness restriction may be removed by appropriate application of results from open shop theory, as indicated in [7], but this is beyond the scope of this paper.

$I^{[m]}$ of the single machine problem. Then

$$Z^{MBD[m]}(I) \doteq w^\top M^{LP[m]} + \frac{1}{2\,s^{[m]}} w^\top p \le Z^*(I) \,. \quad (5)$$

Proof: Let $Z^{[m]}(I)$ denote the optimum value of the minsum objective $\sum_j w_j C^{[m]}$ among all feasible preemptive schedules for instance $I^{[m]}$. From Theorem 2 it follows that $w^\top M^{LP[m]} + \frac{1}{2\,s^{[m]}} w^\top p \le Z^{[m]}(I)$. From the inequalities $C_j^{[m]} \le C_j$ in Lemma 1 and $w \ge 0$, it follows that $Z^{[m]}(I) \le Z^*(I)$. This suffices to prove the corollary. ■

Remark 1: For the problem $P||\sum w_j C_j$ with identical parallel machines and all release dates $r_j = 0$, each $s_i = 1$. Therefore, for any nonpreemptive parallel machines schedule, the mean busy date M_j and completion time C_j of every job j satisfy $M^j = C^j - \frac{1}{2}p_j$. On the other hand $s^{[m]} = m$ and, since the LP schedule is nonpreemptive for identical release dates, $M_j^{LP[m]} = C_j^{LP[m]} - p_j/2m$. Applying the mean busy date relationships $M_j^{[m]} = M_j^*$ of Lemma 1 to the MBD vector $M^* = C^* - \frac{1}{2}p$ of an optimal parallel machine schedule, we obtain the slightly stronger bound: $Z^*(I) = w^\top C^* = w^\top M^* + \frac{1}{2} w^\top p = w^\top M^{[m]} + \frac{1}{2} w^\top p \ge w^\top M^{LP[m]} + \frac{1}{2} w^\top p = w^\top C^{LP[m]} + \frac{1}{2}\left[1 - \frac{1}{m}\right] w^\top p \,.$ We refer to the above inequality as Inequality (1).

Let $Z_n(I) = w^\top p$ denote the optimum value of the n-machine version of the problem, and $Z_1(I) = w^\top \left[M^{LP[m]} + \frac{1}{2m} w^\top p\right]$ the minsum objective value of the LP schedule for instance $I^{[m]}$ of the single speed-m machine version of the problem. Recall that, in the absence of release dates, $Z_1(I)$ is the optimum value of a feasible nonpreemptive schedule on a single machine operating at m times the speed of each given parallel machine. Inequality (1) may be written as

$$Z^*(I) - \frac{1}{2} Z_n(I) \ge \frac{1}{m}\left(Z_1(I) - \frac{1}{2} Z_n(I)\right)$$

which is precisely the lower bound obtained in [2] using algebraic and geometric arguments.

III. Asymptotic Optimality of the WSPR Rule for Uniform Parallel Machines

We now show the asymptotic optimality of the WSPR rule for uniform parallel machines. The simple version of the WSPR heuristic considered herein is defined as follows: whenever a machine becomes idle, start processing the available job, if any, with highest WSPR priority, i.e., job j such that $j < k$ according to (2); if no job is available, wait until the next job release date. Note that we allow the assignment of jobs to machines to be otherwise arbitrary. (We suspect that one can design versions of the uniform parallel machine WSPR heuristic which may be preferable according to some other performance measure, but this is not needed for the present asymptotic analysis.) As before, let C_j^{WSPR} (resp., M_j^{WSPR}) denote the completion time (resp., mean busy date) of job j in the WSPR schedule.[2] Recall that s_m is the speed of the slowest machine and, to simplify, let $p_{max} = \max_{j \in N} p_j$.

Following [2], it is easy to obtain a job-by-job bound for the WSPR schedule in the absence of release dates:

Lemma 2 (Job-by-Job Bound Without Release Dates): For the uniform parallel machines problem $Q||\sum w_j C_j$ without release dates, the completion time vectors of the WSPR and LP schedules satisfy

$$C_j^{WSPR} \le C_j^{LP} + \left(\frac{1}{s_m} - \frac{1}{s^{[m]}}\right) p_{max} \quad \text{for all } j \in N. \quad (6)$$

Proof: Assuming the jobs in WSPR order (2), the completion time of job j in the LP schedule is $C_j^{LP} = p([j])/s^{[m]}$. In the WSPR schedule, job j starts at the earliest completion time of a job in $[j-1]$, that is, no later than $p([j-1])/s^{[m]}$, and completes at most p_j/s_m time units later. Therefore $C_j^{WSPR} \le C_j^{LP} + \left[1/s_m - 1/s^{[m]}\right] p_j$. This implies (6). ■

We now turn to the case with release dates $r_j \ge 0$. Let $N(i)$ denote the set of jobs processed on machine M_i in the WSPR schedule. Since $M_j^{WSPR} = C_j^{WSPR} - \frac{1}{2} w_j p_j/s_i$ for all $j \in N(i)$, we have

$$\begin{aligned} Z^{WSPR} &= w^\top M^{WSPR} + \frac{1}{2}\sum_{i=1}^{m}\sum_{j\in N(i)} w_j \frac{p_j}{s_i} \\ &\le w^\top M^{WSPR} + \frac{1}{2\,s_m} w^\top p \,. \end{aligned} \quad (7)$$

Combining inequalities (5) and (7) with the decomposition (4) of the MBD objective, we only need to compare the processing date functions of the speed-$s^{[m]}$ machine LP schedule and of the parallel machines WSPR schedule. The next Lemma shows that, for any instance with a fixed set of machines, no amount of work from any set $[h]$ can, in the parallel machines WSPR schedule, be delayed, relative to the single machine LP schedule, by more than a constant multiple of p_{max} time units.

Lemma 3 (Parallel Machines Work Delay Lemma): Assume the jobs are ranked according to the WSPR order (2). Consider the WSPR schedule on uniform parallel machines, and the speed-$s^{[m]}$ machine LP schedule defined above. Then, for all $h \le n$ and for all $0 < q \le p([h])$,

$$\bar{R}_{[h]}^{WSPR}(q) \le \bar{R}_{[h]}^{LP}(q) + \left(\frac{1}{s_1} + \frac{m-1}{s_m} + \frac{s^{[m]}}{(s_m)^2}\right) p_{max} \quad (8)$$

Proof: We fix $h \in \{1, \ldots, n\}$ and we define

$$\alpha = m - 1 + \frac{s^{[m]}}{s_m} \,.$$

[2] To properly speak of “the WSPR schedule” we would need to define a rule for assigning jobs to machines in case several machines are available when a job starts processing. For example, we may assign the highest priority available job to a fastest available machine. In fact, our analysis applies to any nonpreemptive feasible schedule which is consistent with the stated WSPR priority rule, irrespective of the details of such machine assignments.

We start by considering the LP schedule on the speed-$s^{[m]}$ machine. Let $[a_k; b_k]$ (where $k = 1;\ldots;K$) denote the disjoint time intervals during which set $[h]$ is being processed continuously in the LP schedule. Thus $0 \le a_1$ and $b_{k-1} < a_k$ for $k = 2;\ldots;K$. The unprocessed work function $R^{LP}_{[h]}$ starts with $R^{LP}_{[h]}(t) = p([h])$ for $0 \le t \le a_1$; decreases at rate $s^{[m]}$ in the intervals $[a_k; b_k]$ while remaining constant outside these intervals; and it ends with $R^{LP}_{[h]}(t) = 0$ for $b_K \le t \le T$. Let $J(k) = \{j \in [h] : a_k < C^{LP}_j \le b_k\}$ denote the set of jobs in $[h]$ that are processed during time interval $[a_k; b_k]$ in the LP schedule. Note that $a_k = \min_{j \in J(k)} r_j$ and $b_k = a_k + p(J(k))/s^{[m]}$. Furthermore, $Q_k = \sum_{\ell > k} p(J(\ell))$ is the total work from set $[h]$ released after date b_k, where $Q_K = 0$ and $Q_0 = p([h])$. For all $k = 1;\ldots;K$ we have $Q_k = Q_{k-1} - p(J(k))$. In the interval $[Q_k; Q_{k-1})$ the processing date function $\bar{R}^{LP}_{[h]}$ decreases at rate $1/s^{[m]}$ from $\bar{R}^{LP}_{[h]}(Q_k) = b_k$. Thus $\bar{R}^{LP}_{[h]}(q) = a_k + (Q_{k-1} - q)/s^{[m]}$ for all $Q_k \le q < Q_{k-1}$.

Now consider the WSPR schedule on the uniform parallel machines and fix an interval $[a_k; b_k)$. We claim that, for every $k = 1;\ldots;K$ and every date $a_k \le t < b_k$ the unprocessed work

$$R^{WSPR}_{[h]}(t) \le R^{LP}_{[h]}(t) + \alpha p_{max} : \tag{9}$$

By contradiction, assume that (9) is violated at date $t \in [a_k; b_k)$. Let $\hat{t} = \inf\{t : (9) \text{ is violated}\}$. Since the functions $R^{WSPR}_{[h]}$ and $R^{LP}_{[h]}(t)$ are continuous, $R^{WSPR}_{[h]}(\hat{t}) \ge R^{LP}_{[h]}(\hat{t}) + \alpha p_{max}$, and the difference $R^{WSPR}_{[h]}(t) - R^{LP}_{[h]}(t)$ is strictly increasing immediately to the right of $\hat{t}$. But since $R^{LP}_{[h]}$ is constant outside the intervals $[a_k; b_k]$ and $R^{WSPR}_{[h]}$ then we must have $a_k \le \hat{t} < b_k$ for some $k \in \{1;\ldots;K\}$. This implies that at least one machine M_i is not processing a job in $[h]$ immediately after date $\hat{t}$. If at least one machine is idle just after date $\hat{t}$ then let $\theta = \hat{t}$; otherwise, let $\theta \le \hat{t}$ be the latest start date of a job not in $[h]$ and in process just after date $\hat{t}$. Since no job in $[h]$ was available for processing at date θ, then all work released no later than θ must either have been completed, or be started on a machine $M_u \ne M_i$. Note that at least $p([h]) - R^{LP}_{[h]}(\theta)$ units of work have been released by date θ. On the other hand, a total of at most $(m-1)p_{max}$ units of work can be started on machines $M_u \ne M_i$ just after date $\hat{t}$. Therefore

$$p([h]) - R^{LP}_{[h]}(\theta) \le p([h]) - R^{WSPR}_{[h]}(\theta) + (m-1)p_{max}$$

If $\theta < \hat{t}$ then the job $j \notin [h]$ started at date θ has processing requirement $p_j \le p_{max}$ and is processed at least at the slowest machine speed s_m. Since this job is still in process at date $\hat{t}$, we must have $\hat{t} < \theta + p_{max}/s_m$. The unprocessed work function $R^{LP}_{[h]}$ decreases by at most $s^{[m]}(\hat{t} - \theta)$ between dates θ and $\hat{t}$, whereas $R^{WSPR}_{[h]}$ is nonincreasing. Therefore

$$\begin{aligned} R^{LP}_{[h]}(\hat{t}) &\ge R^{LP}_{[h]}(\theta) - s^{[m]}(\hat{t} - \theta) \\ &> R^{LP}_{[h]}(\theta) - s^{[m]}\frac{p_{max}}{s_m} \\ &\ge R^{WSPR}_{[h]}(\theta) - (m-1)p_{max} - s^{[m]}\frac{p_{max}}{s_m} \\ &\ge R^{WSPR}_{[h]}(\hat{t}) - \alpha p_{max} \\ &\ge R^{LP}_{[h]}(\hat{t})\,; \end{aligned}$$

a contradiction. Thus claim (9) is proved.

Claim (9) implies that whenever $Q_k + \alpha p_{max} < q < Q_{k-1}$, the processing date functions satisfy

$$\bar{R}^{WSPR}_{[h]}(q) \le \bar{R}_{[h]}(q) + \frac{\alpha p_{max}}{s^{[m]}} : \tag{10}$$

Let $\hat{q} = \min\{Q_k + \alpha p_{max}\ ;\ Q_{k-1}\}$ and consider the last $\hat{q}$ units of work released from set $J(k)$. If $\hat{q} < Q_{k-1}$ then claim (9) implies that $\bar{R}^{WSPR}_{[h]}(\hat{q}) \le b_k$, that is, the first $p([h]) - \hat{q}$ units of work are completed by date b_k. If some of the remaining $\hat{q}$ units of work is being processed at a date $\hat{t} > b_k$ then, since all work from $J(k)$ has been released by date b_k, there will be no date τ at which no work from $J(k)$ is in process until all this work from $J(k)$ is completed, that is, until date $\bar{R}^{WSPR}_{[h]}(Q_k)$. Furthermore, this work is processed at least at the minimum speed $s_m > 0$, so $\bar{R}^{WSPR}_{[h]}(Q_k) \le \hat{t} + \hat{q}/s_m$. Note also that, unless $\bar{R}^{WSPR}_{[h]}(Q_k) = b_k$, a machine becomes available for processing these $\hat{q}$ units of work between dates b_k and $b_k + p_{max}/s_1$, where s_1 is the fastest speed of a machine. Thus, for $Q_k \le q \le Q_k + \hat{q}$ we have

$$\bar{R}^{LP}_{[h]}(q) = b_k - (q - Q_k)/s^{[m]} \tag{11}$$

and

$$\bar{R}^{WSPR}_{[h]}(q) \le b_k + \frac{p_{max}}{s_1} + \frac{\alpha p_{max} - q}{s_m} : \tag{12}$$

Inequalities (10), (11) and (12) imply (8) and the proof is complete. ∎

Integrating inequality (8) from 0 to $p([h])$ implies

$$\begin{aligned} &\int_0^{p([h])} \left(\bar{R}^{WSPR}_{[h]}(q) - \bar{R}^{LP}_{[h]}(q) \right) dq \\ &\le \left(\frac{1}{s_1} + \frac{m-1}{s_m} + \frac{s^{[m]}}{(s_m)^2} \right) p_{max}\, p([h]) : \end{aligned} \tag{13}$$

The next theorem combines inequality (13) with the cost decomposition (4) and inequalities (5) and (7), to derive a $O(n)$ bound on the difference between the minsum objective values of the parallel machines WSPR schedule and the single machine LP schedule, for all instances with bounded weights and processing requirements.

Theorem 3: Consider any instance of the uniform parallel machine problem $Q|r_j|\sum w_j C_j$ such that $0 \le w_j \le \bar{w}$ and $0 < \underline{p} \le p_j \le \bar{p}$ for all jobs $j \in N$. Then

$$Z^{WSPR}(I) \le Z^{MBD[m]}(I) + \beta \bar{w} \bar{p} n$$

$$\text{where} \quad \beta = \frac{\bar{p}}{\underline{p}} \left(\frac{1}{s_1} + \frac{m-1}{s_m} + \frac{s^{[m]}}{(s_m)^2} \right) + \frac{1}{2} \left(\frac{1}{s_m} - \frac{1}{s^{[m]}} \right) : \tag{14}$$

Proof: Using (4), inequality (13), all $\Delta_h \geq 0$, and the given bounds on all w_j and p_j, we have

$$
\begin{aligned}
& Z^{WSPR}(I) - Z^{MBD[m]}(I) \\
\leq\ & w^{\top} M^{WSPR} + \frac{1}{2s_m} w^{\top} p \\
-\ & \left(w^{\top} M^{LP[m]} + \frac{1}{2s^{[m]}} w^{\top} p \right) \\
=\ & \sum_{h=1} \Delta_h \int_0^{p([h])} \left(\bar{R}^{WSPR}_{[h]}(q) - \bar{R}^{LP}_{[h]}(q) \right) dq \\
+\ & \left(\frac{1}{2s_m} - \frac{1}{2s^{[m]}} \right) w^{\top} p \\
\leq\ & \sum_{h=1} \Delta_h \left(\frac{1}{s_1} + \frac{m-1}{s_m} + \frac{s^{[m]}}{(s_m)^2} \right) p_{max}\, p([h]) \\
+\ & \frac{1}{2} \left(\frac{1}{s_m} - \frac{1}{s^{[m]}} \right) n \bar{w} \bar{p} \\
\leq\ & \frac{w_1}{p_1} \left(\frac{1}{s_1} + \frac{m-1}{s_m} + \frac{s^{[m]}}{(s_m)^2} \right) n \bar{p}^2 \\
+\ & \frac{1}{2} \left(\frac{1}{s_m} - \frac{1}{s^{[m]}} \right) n \bar{w} \bar{p} \\
\leq\ & n \bar{w} \bar{p} \left(\frac{\bar{p}}{\underline{p}} \left(\frac{1}{s_1} + \frac{m-1}{s_m} + \frac{s^{[m]}}{(s_m)^2} \right) + \frac{1}{2} \left(\frac{1}{s_m} - \frac{1}{s^{[m]}} \right) \right)
\end{aligned}
$$

This proves Theorem 3. ■

Now we are ready to prove Theorem 1.

Proof: [of Theorem 1] For every instance $I \in \mathcal{I}$, let $Z^*(I)$ (resp., $Z^{MBD}(I)$; resp., $Z^{WSPR}(I)$) denote the minsum objective of an optimal non-preemptive schedule (resp., the LP schedule; resp., a WSPR schedule). Theorem 2 and Theorem 3 imply

$$Z^{MBD[m]}(I) \leq Z^*(I) \leq Z^{WSPR}(I) \leq Z^{MBD[m]}(I) + \beta \bar{w} \bar{p} n;$$

where β is as defined in (14).

Note that $Z^{MBD[m]}(I) \geq \frac{\underline{w}}{s^{[m]}} \frac{n(n+1)}{2} \underline{p}$. Therefore

$$\frac{Z^{WSPR}(I)}{Z^*(I)} \leq 1 + \frac{2s^{[m]}}{n+1} \frac{\bar{w}\bar{p}}{\underline{w}\,\underline{p}}.$$

Thus, for every $r > 1$, there exists n_0 such that for all instances $I \in \mathcal{I}$ with $n \geq n_0$ we have $Z^{WSPR}(I)/Z^*(I) \leq r$. The proof is complete. ■

Zero processing time jobs: Assume now that we have a set Z, disjoint from N, of zero jobs j with $p_j = 0$. The total number of jobs is now $n' = n + |Z|$. Note that, for all $j \in Z$, $C_j^{LP} = r_j$ since every job $j \in Z$ is immediately inserted into the LP schedule at date r_j. On the other hand, $C_j^{WSPR} < r_j + p_{max}/s_1$, where $p_{max} = \max_{j \in N} p_j \leq \bar{p}$ denotes the longest processing requirement, and s_1 is the fastest machine speed; indeed, in the WSPR schedule every job $j \in Z$ is processed either at date r_j or else at the earliest completion of a job in N in process at date r_j. Therefore, with β as defined in Theorem 1 and assuming $w_j \leq \bar{w}$ for all $j \in Z$, we have

$$\sum_{j \in N \cup Z} w_j C_j^{WSPR} - \sum_{j \in N \cup Z} w_j C_j^{LP} \leq \beta \bar{w} \bar{p} n + w(Z) \frac{\bar{p}}{s_1} \leq \bar{w}\bar{p} \beta n'$$

since $\beta \geq 1/s_1$. So the $O(n')$ bound in Theorem 1 extends to the case of zero processing time jobs if one defines $\underline{p} = \min\{p_j : p_j > 0\}$.

For Theorem 3 to extend to this case as well, it suffices that the number n of nonzero jobs grow faster than the square root $\sqrt{n'}$ of the total number of jobs. Indeed in such a case a lower bound on the MBD objective value $Z^{MBD}(I)$, which is quadratic in n, grows faster than linearly in n'. In this respect, one may recall the class of “bad instances” presented in [4] for the single machine problem, which we rescale here by dividing processing times by n' and multiplying weights by n', so $\bar{p} = \bar{w} = 1$. For these instances the objective value $Z^{WSPR}(I)$ approaches $e \approx 2.718$ whereas the MBD lower bound $Z^{MBD}(I)$ approaches $e - 1$. Thus one cannot use this MBD lower bound to establish the asymptotic optimality of the WSPR schedule in this case. This is due to the fact that, for these instances, the number of nonzero jobs is in fact constant (equal to one), and the optimum objective value does not grow at a faster rate than the additive error bound.

References

[1] Chekuri, C., Motwani, R., Natarajan, B., Stein, C.: Approximation Techniques for Average Completion Time Scheduling. Proceedings of the Eight Annual ACM-SIAM Symposium on Discrete Algorithms (1997) 609–618

[2] Eastman, W. L., Even, S., Isaacs, I. M.: Bounds for the Optimal Scheduling of n Jobs on m Processors. Management Science 11 (1964) 268–279

[3] Goemans, M. X.: Improved Approximation Algorithms for Scheduling with Release Dates. Proceedings of the 8th ACM-SIAM Symposium on Discrete Algorithms (1997) 591-598

[4] Goemans, M. X., Queyranne, M., Schulz, A. S., Skutella, M., Wang, Y.: Single Machine Scheduling with Release Dates. Report 654, Fachbereich Mathematik (1999), Technische Universität Berlin, Germany. Available at URL: http://www.math.tu-berlin.de/coga/publications/techreports/1999/Report-654-1999.html

[5] Graham, R.L., Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H. G.: Optimization and approximation in deterministic sequencing and scheduling: a survey. Annals of Discrete Mathematics 5 (1979) 287–326

[6] Kaminsky, P., Simchi-Levi, D.: Probabilistic Analysis of an On-Line Algorithm for the Single Machine Completion Time Problem With Release Dates. Operations Research Letters 21 (2001) 141–148

[7] Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H. G., Shmoys, D. B.: Sequencing and Scheduling: Algorithms and Complexity. In: S. C. Graves, A. H. G. Rinnooy Kan and P. H. Zipkin (eds.), Logistics of Production and Inventory, Handbooks in Operations Research and Management Science 4 (1993), North–Holland, Amsterdam.

[8] Lenstra, J. K., Rinnooy Kan, A. H. G., Brucker, P.: Complexity of Machine Scheduling Problems. Annals of Discrete Math 1 (1977) 343–362

[9] Queyranne, M., Sviridenko, M.: Approximation algorithms for shop scheduling problems with minsum objective. Faculty of Commerce, University of British Columbia (1999)

[10] Sgall, J.: On-line scheduling | a survey. In: A. Fiat and G.J. Woeginger (eds.), Online Algorithms: The State of the Art, Lecture Notes in Computer Science 1442 (1998) 196{231, Springer, Berlin.

[11] Smith, W.: Various optimizers for single-stage production. Naval Res. Logist. Quart. 3 (1956) 59{66

[12] Uma, R. N., Wein, J.: On the Relationship between Combinatorial and LP-Based Approaches to NP-hard Scheduling Problems. In: R. E. Bixby, E. A. Boyd and R. Z. Rios-Mercado (eds.), Integer Programming and Combinatorial Optimization. Proceedings of the Sixth International IPCO Conference, Lecture Notes in Computer Science 1412 (1998) 394{408, Springer, Berlin.